Polynomial time cryptanalysis of noncommutative-algebraic key exchange protocols

Boaz Tsaban

Department of Mathematics, Bar-Ilan University, Ramat Gan 52900, Israel
tsaban@math.biu.ac.il
http://www.cs.biu.ac.il/~tsaban

arXiv:1210.8114v1 [cs.CR] 30 Oct 2012

Abstract. We introduce the *linear centralizer method* for a passive adversary to extract the shared key in group-theory based key exchange protocols (KEPs). We apply this method to obtain a polynomial time cryptanalysis of the *Commutator KEP*, introduced by Anshel–Anshel–Goldfeld in 1999 and considered extensively ever since. We also apply this method to the *Centralizer KEP*, introduced by Shpilrain–Ushakov in 2006. Our method is proved to be of polynomial time using a technical lemma about sampling invertible matrices from a linear space of matrices.

1 Introduction

Key Exchange Protocols (KEPs) make it possible for two electronic entities, *Alice* and *Bob*, to establish a shared secret key over a public communication channel. Since Diffie and Hellman's 1976 breakthrough KEP, few alternative KEP proposals resisted cryptanalysis. This, together with the (presently, theoretical) issue that the Diffie–Hellman and other classic KEPs can be broken in polynomial time by quantum computers, is a strong motivation for searching for substantially different KEPs. Lattice-based KEPs [29] seem to be a viable potential alternative. Both the classic KEPs and the Lattice-based ones are based on commutative algebraic structures.

In 1999, Anshel, Anshel, and Goldfeld [2] (cf. [3]) introduced the *Commutator KEP*, a general method for constructing KEPs based on *noncommutative* algebraic structures. Around the same time, Ko, Lee, Cheon, Han, Kang, and Park [19] introduced the *Braid Diffie–Hellman KEP*, another general method achieving the same goal. The security of both KEPs is based on variations of the *Conjugacy Search Problem (CSP)*: Given conjugate elements g, h in a noncommutative group, find x in that group such that $x^{-1}gx = h$. Both papers [2] and [19] proposed to use the *braid group* $\mathbf{B}_N$, a finitely presented, infinite noncommutative group parameterized by a natural number N, as the platform group.

The introduction of the Commutator KEP and the Braid Diffie–Hellman KEP was followed by a stream of heuristic attacks (e.g., [14], [15], [22], [13], [9], [24], [10], [11], [16], [23], [25], [27], [28]),[1] demonstrating that the weak keys of these KEPs constitute a substantial portion of the key space, or more precisely, that the two most natural distributions on the braid group $\mathbf{B}_N$ seem to give rise to insecure KEPs. Consequently, a program was set forth, by several independent research groups, to find an efficient way to sample hard instances of the underlying computational problems (e.g., [23], [20], [12], [1]).

Most of the mentioned heuristic attacks address the Commutator KEP, and not the Braid Diffie–Hellman KEP. The reason is that in 2003, Cheon and Jun published a polynomial time cryptanalysis of the Braid Diffie–Hellman KEP, using an ingenious representation theoretic

[1] Surveys of some of the heuristic attacks are provided in Dehornoy [7] and Garber [8].

method [6]. In their paper, Cheon and Jun stress that their cryptanalysis does not apply directly to the Commutator KEP. Thus far, no polynomial time attack was found on the Commutator KEP, whose success does not depend on the distributions used to generate the random group elements. The main result of the present paper is a Las Vegas, provable polynomial time cryptanalysis of the Commutator KEP [2], in the passive adversary model, that succeeds regardless of the distributions used to generate the keys. This cryptanalysis constitutes a polynomial time solution of the underlying computational problem.

The methods developed for the new cryptanalysis are applicable to additional KEPs in the context of group-based cryptography. We present an application of these methods to the *Centralizer KEP*, introduced by Shpilrain and Ushakov in 2006 [33], to obtain a polynomial time attack. This is the first cryptanalysis, of any kind, of the Centralizer KEP.

We stress that the Cheon and Jun cryptanalysis and the ones presented here, while of polynomial time, are impractical for standard values of N (e.g., $N = 100$). These results are of theoretic nature. Ignoring logarithmic factors, the complexity of our cryptanalyses is about N^{17}, times a cubic polynomial in the other relevant parameters. Incidentally, though, these cryptanalyses establish the first provable practical attacks in the case where the index N of the braid group $\mathbf{B}_N$ is small, e.g., when $N = 8$.

Section 2 introduces the Commutator KEP and the braid group. In Section 3, we eliminate a technical complexity theoretic obstacle. Section 4 applies a method of Cheon and Jun to reduce our problem to matrix groups over finite fields. Section 5 is the main ingredient of our cryptanalysis, cryptanalyzing the Commutator KEP in matrix groups. Section 6 fills a gap in our proof, by applying the Schwartz–Zippel Lemma to obtain a lower bound on the probability that certain random matrices are invertible. Section 7 is a cryptanalysis of the Centralizer KEP, using the methods introduced in the earlier sections. Some final comments and additional applications are provided in Section 8.

2 The Commutator KEP and the braid group $\mathbf{B}_N$

We will use, throughout, the following basic notation.

Notation 1 *For a noncommutative group G and group elements $g, x \in G$, $g^x := x^{-1}gx$, the conjugate of g by x.*

Useful identities involving this notation, that are easy to verify, include $g^{xy} = (g^x)^y$, and $g^c = g$ for every *central* element $c \in G$, that is, such that $ch = hc$ for all $h \in G$.

The *Commutator KEP* [2] is described succinctly in Figure 1.[2] In some detail:

1. A noncommutative group G and elements $a_1, \ldots, a_k, b_1, \ldots, b_k \in G$ are publicly given.[3]
2. Alice and Bob choose free group words in the variables $x_1, \ldots, x_k$, $v(x_1, \ldots, x_k)$ and $w(x_1, \ldots, x_k)$, respectively.[4]
3. Alice substitutes $a_1, \ldots, a_k$ for $x_1, \ldots, x_k$, to obtain a secret element $a = v(a_1, \ldots, a_k) \in G$. Similarly, Bob computes $b = w(b_1, \ldots, b_k) \in G$.

[2] In our diagrams, green letters indicate publicly known elements, and red ones indicate secret elements, known only to the secret holders. Results of computations involving elements of both colors may be either publicly known, or secret, depending on the context. The colors are not necessary to follow the diagrams.

[3] By adding elements, if needed, we assume that the number of a_i's is equal to the number of b_i's.

[4] A free group word in the variables $x_1, \ldots, x_k$ is a product of the form $x_{i_1}^{\epsilon_1} x_{i_2}^{\epsilon_2} \cdots x_{i_m}^{\epsilon_m}$, with $i_1, \ldots, i_m \in \{1, \ldots, k\}$ and $\epsilon_1, \ldots, \epsilon_m \in \{1, -1\}$, and with no subproduct of the form $x_i x_i^{-1}$ or $x_i^{-1} x_i$.

4. Alice sends the conjugated elements ${b_1}^a, \ldots, {b_k}^a$ to Bob, and Bob sends ${a_1}^b, \ldots, {a_k}^b$ to Alice.
5. The shared key is the *commutator* $a^{-1}b^{-1}ab$.

As conjugation is a group isomorphism, we have that

$$v({a_1}^b, \ldots, {a_k}^b) = v(a_1, \ldots, a_k)^b = a^b = b^{-1}ab.$$

Thus, Alice can compute the shared key $a^{-1}b^{-1}ab$ as $a^{-1}v({a_1}^b, \ldots, {a_k}^b)$, using her secret $a, v(x_1, \ldots, x_k)$ and the public elements ${a_1}^b, \ldots, {a_k}^b$. Similarly, Bob computes $a^{-1}b^{-1}ab$ as $w({b_1}^a, \ldots, {b_k}^a)^{-1}b$.

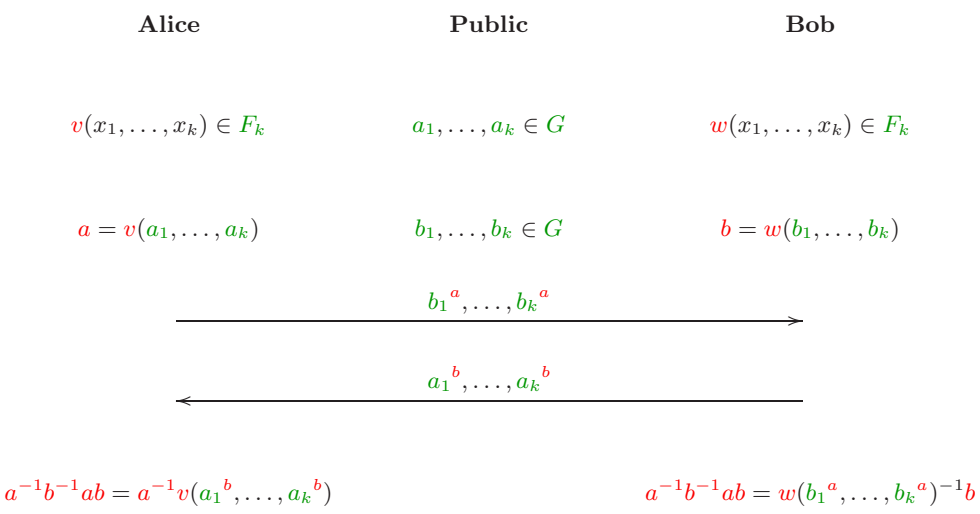

Fig. 1. The Commutator KEP

For the platform group G, it is proposed in [2] to use the *braid group* $\mathbf{B}_N$, a group parameterized by a natural number N. The interested reader will find detailed information on $\mathbf{B}_N$ in almost each of the papers in the bibliography. We quote here the information needed for the present paper.

Let S_N be the symmetric group of permutations on N symbols. For our purposes, the braid group $\mathbf{B}_N$ is a group of elements of the form

$$(i, \mathbf{p}),$$

where i is an integer, and $\mathbf{p}$ is a finite (possibly, empty) sequence of elements of S_N, that is, $\mathbf{p} = (p_1, \ldots, p_\ell)$ for some $\ell \geq 0$ and $p_1, \ldots, p_\ell \in S_N$. The sequence $\mathbf{p} = (p_1, \ldots, p_\ell)$ is requested to be *left weighted* (a property whose definition will not be used here), and p_1 must not be the involution $p(k) = n - k$.[5]

Elements of $\mathbf{B}_N$ are called *braids*, for they may be identified with braids on N strands. This identification, however, will play no role in the present paper. For "generic" braids $(i, (p_1, \ldots, p_\ell)) \in \mathbf{B}_N$, i is negative and $|i|$ is $O(\ell)$, but this is not always the case. Note that the bitlength of an element $(i, (p_1, \ldots, p_\ell)) \in \mathbf{B}_N$ is $O(\log |i| + \ell N \log N)$.

[5] For readers familiar with the braid group, we point out that the sequence $(i, (p_1, \ldots, p_\ell))$ codes the left normal form $\Delta^i p_1 \cdots p_\ell$ of the braid, in Artin's presentation, with Δ being the fundamental, full twist braid.

Multiplication is defined on $\mathbf{B}_N$ by an algorithm of complexity $O(\ell^2 N \log N + \log|i|)$. Inversion is of linear complexity. Explicit implementations are provided, for example, in [5].

For a passive adversary to extract the shared key of the Commutator KEP out of the public information, it suffices to solve the following problem.

Problem 2 (Commutator KEP Problem) *Let $a_1, \dots, a_k, b_1, \dots, b_k \in \mathbf{B}_N$, each of the form $(i, \mathbf{p})$ with $\mathbf{p}$ of length $\le \ell$. Let a be a product of at most m elements of $\{a_1, \dots, a_k\}^{\pm 1}$, and let b be a product of at most m elements of $\{b_1, \dots, b_k\}^{\pm 1}$.*
Given $a_1, \dots, a_k, b_1, \dots, b_k, a_1^b, \dots, a_k^b, b_1^a, \dots, b_k^a$, compute $a^{-1}b^{-1}ab$.

Our solution of Problem 2 consists of several ingredients.

3 Reducing the infimum

The *infimum* of a braid $b = (i, \mathbf{p})$ is the integer $\inf(b) := i$. As the bitlength of b is $O(\log|i| + \ell N \log N)$, an algorithm polynomial in $|i|$ would be at least *exponential* in the bitlength. We first remove this obstacle.

In cases where $\mathbf{p}$ is the empty sequence, we write (i) instead of $(i, \mathbf{p})$. The properties of $\mathbf{B}_N$ include, among others, the following ones.

(a) $(i) \cdot (j, \mathbf{p}) = (i + j, \mathbf{p})$ for all integers i and all $(j, \mathbf{p}) \in \mathbf{B}_N$.
In particular, $(i) = (1)^i$ for all i.
(b) $(2) \cdot (i, \mathbf{p}) = (i, \mathbf{p}) \cdot (2)$ for all for all $(i, \mathbf{p}) \in \mathbf{B}_N$.

Thus, $(2j)$ is a central element of $\mathbf{B}_N$ for each integer j. If follows that, for each $(i, \mathbf{p}) \in \mathbf{B}_N$,

$$(i, \mathbf{p}) = (i - (i \bmod 2)) \cdot (i \bmod 2, \mathbf{p}).$$

This way, every braid $b \in \mathbf{B}_N$ decomposes to a product $c\tilde{b}$, where c is of the form $(2j)$ (and thus *central*), and $\inf(\tilde{b}) \in \{0, 1\}$.

Consider the public information in Figure 1. For each $j = 1, \dots, k$, decompose as above

$$a_j = c_j \tilde{a}_j,$$
$$b_j = d_j \tilde{b}_j,$$

with c_j, d_j central and $\inf(\tilde{a}_j), \inf(\tilde{b}_j) \in \{0, 1\}$ for all $j = 1, \dots, k$. Let

$$\tilde{a} = v(\tilde{a}_1, \dots, \tilde{a}_k);$$
$$\tilde{b} = w(\tilde{b}_1, \dots, \tilde{b}_k);$$
$$c = v(c_1, \dots, c_k);$$
$$d = w(d_1, \dots, \tilde{d}_k).$$

As the elements c_j, d_j are central, we have that

$$\tilde{a} = v(c_1^{-1}a_1, \dots, c_k^{-1}a_k) = v(c_1^{-1}, \dots, c_k^{-1}) \cdot v(a_1, \dots, a_k) = c^{-1}a.$$

Similarly, $\tilde{b} = d^{-1}b$. As c and d are central,

$$a_j^b = (c_j \tilde{a}_j)^b = c_j \tilde{a}_j^b = c_j \tilde{a}_j^{d\tilde{b}} = c_j \tilde{a}_j^{\tilde{b}}$$

for all $j = 1, \ldots, k$. Thus, $\tilde{a}_j^{\tilde{b}}$ can be computed for all j. Similarly, $\tilde{b}_j^{\tilde{a}}$ can be computed. Now,

$$a^{-1}b^{-1}ab = (c\tilde{a})^{-1}(d\tilde{b})^{-1}(c\tilde{a})(d\tilde{b}) = \tilde{a}^{-1}c^{-1}\tilde{b}^{-1}d^{-1}c\tilde{a}d\tilde{b} = \tilde{a}^{-1}\tilde{b}^{-1}\tilde{a}\tilde{b}.$$

This shows that the Commutator KEP Problem is reducible, in linear time, to the same problem using $\tilde{a}_1, \ldots, \tilde{a}_k, \tilde{b}_1, \ldots, \tilde{b}_k$ instead of $a_1, \ldots, a_k, b_1, \ldots, b_k$. Thus, we may assume that

$$\inf(a_1), \ldots, \inf(a_k), \inf(b_1), \ldots, \inf(b_k) \in \{0, 1\}$$

to start with. Assume that henceforth.

For a braid $x = (i, \mathbf{p})$, let $\ell(\mathbf{p})$ be the number of permutations in the sequence $\mathbf{p}$. For integers i, s, let

$$[i, s] = \{x \in \mathbf{B}_N \,:\, i \leq \inf(x) \leq \inf(x) + \ell(x) \leq s\}.$$

We use the following basic facts about $\mathbf{B}_N$:

1. If $x_1 \in [i_1, s_1]$ and $x_2 \in [i_2, s_2]$, then $x_1x_2 \in [i_1 + i_2, s_1 + s_2]$.
2. If $x \in [i, s]$, then $x^{-1} \in [-i - s, -i]$.

Thus, for each $x \in \{a_1, \ldots, a_k, b_1, \ldots, b_k\}^{\pm 1}$, $x^{\pm 1} \in [-\ell - 1, \ell + 1]$, and therefore, in the notation of our problem, $a, b \in [-m(\ell + 1), m(\ell + 1)]$. Thus,

$$a^{-1}b^{-1}ab \in [-4m(\ell + 1), 4m(\ell + 1)].$$

Corollary 3 *In the Commutator KEP Problem, $a^{-1}b^{-1}ab \in [-4m(\ell + 1), 4m(\ell + 1)]$.*

4 Reducing to a matrix group over a finite field

Let n be a natural number. As usual, we denote the algebra of all $n \times n$ matrices over a field $\mathbb{F}$ by $\mathrm{M}_n(\mathbb{F})$, and the group of invertible elements of this algebra by $\mathrm{GL}_n(\mathbb{F})$. A *matrix group* is a subgroup of $\mathrm{GL}_n(\mathbb{F})$. A *faithful representation* of a group G in $\mathrm{GL}_n(\mathbb{F})$ is a group isomorphism from G onto a matrix group $H \leq \mathrm{GL}_n(\mathbb{F})$. A group is *linear* if it has a faithful representation.

Bigelow and, independently, Krammer, established in their breakthrough papers [4], [21] that the braid group $\mathbf{B}_N$ is linear, by proving that the so-called *Lawrence–Krammer representation*

$$\mathrm{LK}\colon \mathbf{B}_N \longrightarrow \mathrm{GL}_{\binom{N}{2}}(\mathbb{Z}[t^{\pm 1}, \frac{1}{2}]),$$

whose dimension is

$$n := \binom{N}{2},$$

is injective.[6] The Lawrence–Krammer representation of a braid can be computed very efficiently.[7] It is proved implicitly in [21], and explicitly in [6], that this representation is also invertible in polynomial time. The following result follows from Corollary 1 of [6].

Theorem 4 (Cheon–Jun [6]) *Let $x \in [i, s]$ in $\mathbf{B}_N$. Let $M \geq \max(-i, s)$. Then:*

[6] Bigelow proved this theorem for the coefficient ring $\mathbb{Z}[t^{\pm 1}, q^{\pm 1}]$ with two variables. Krammer proved, in additiona, that one may replace q by any real number from the interval $(0, 1)$.

[7] When the infimum i is polynomial in the other parameters, which we proved in Section 3 that we may assume.

1. *The degrees of* t *in* $\mathrm{LK}(x) \in \mathrm{GL}_n(\mathbb{Z}[t^{\pm 1}, \frac{1}{2}])$ *are in* $\{-M, -M+1, \ldots, M\}$.
2. *The rational coefficients* $\frac{c}{2^d}$ *in* $\mathrm{LK}(x)$ *(*c *integer,* d *nonnegative integer) satisfy:* $|c| \leq 2^{N^2M}, |d| \leq 2NM$.

In the notation of Theorem 4, Theorem 2 in Cheon–Jun [6] implies that inversion of $\mathrm{LK}(x)$ is of order $N^6 \log M$ multiplications of entries. Ignoring logarithmic factors and thus assuming that each entry multiplication costs $NM \cdot N^2M = N^3M^2$, this accumulates to N^8M^2. This complexity is dominated by the complexity of the linear centralizer step of our cryptanalysis (Section 5).

Let us return to the Commutator KEP Problem 2. By Corollary 3,

$$K := a^{-1}b^{-1}ab \in [-4m(\ell+1), 4m(\ell+1)].$$

Let $M = 4m(\ell+1)$. By Theorem 4, we have that

$$(2^{2NM}t^M) \cdot \mathrm{LK}(K) \in \mathrm{GL}_n(\mathbb{Z}[t]),$$

the absolute values of the coefficients in this matrix are bounded by $2^{N^2(M+1)}$, and the maximal degree of t in this matrix is bounded by $2M$.

Let p be a prime slightly greater than 2^{N^2M}, and $f(t)$ be an irreducible polynomial over $\mathbb{Z}_p$, of degree d slightly larger than $2M$. Then

$$(2^{2NM}t^M) \cdot \mathrm{LK}(K) = (2^{2NM}t^M) \cdot \mathrm{LK}(K) \bmod (p, f(t)) \in \mathrm{GL}_n(\mathbb{Z}[t]/\langle p, f(t)\rangle),$$

under the natural identification of $\{-(p-1)/2, \ldots, (p-1)/2\}$ with $\{0, \ldots, p-1\}$.

Let $\mathbb{F} = \mathbb{Z}[t]/\langle p, f(t)\rangle = \mathbb{Z}[t^{\pm 1}, \frac{1}{2}]/\langle p, f(t)\rangle$. $\mathbb{F}$ is a finite field of cardinality p^d, where d is the degree of $f(t)$. It follows that the complexity of field operations in $\mathbb{F}$ is, up to logarithmic factors, of order

$$d^2 \log p = O(M^3N^2) = O(m^3\ell^3N^2).$$

Thus, the key K can be recovered as follows:

1. Apply the composed function $\mathrm{LK}(x) \bmod (p, f(t))$ to the input of the Commutator KEP Problem, to obtain a version of this problem in $\mathrm{GL}_n(\mathbb{F})$.
2. Solve the problem there, to obtain $\mathrm{LK}(K) \bmod (p, f(t))$.
3. Compute $(2^{2NM}t^M) \cdot \mathrm{LK}(K) \bmod (p, f(t)) = (2^{2NM}t^M) \cdot \mathrm{LK}(K)$.
4. Divide by $(2^{2NM}t^M)$ to obtain $\mathrm{LK}(K)$.
5. Compute K using the Cheon–Jun inversion algorithm.

It remains to devise a polynomial time solution of the Commutator KEP Problem in matrix groups.

5 Linear centralizers

In this section, we solve the Commutator KEP Problem in matrix groups. We first state the problem in a general form. As usual, for a group G and elements $g_1, \ldots, g_k \in G$, $\langle g_1, \ldots, g_k\rangle$ denotes the subgroup of G generated by $g_1, \ldots, g_k$.

Problem 5 (Commutator KEP Problem) *Let* G *be a group. Let* $a_1, \ldots, a_k, b_1, \ldots, b_k \in G$. *Let* $a \in \langle a_1, \ldots, a_k\rangle, b \in \langle b_1, \ldots, b_k\rangle$.
Given $a_1, \ldots, a_k, b_1, \ldots, b_k, a_1^b, \ldots, a_k^b, b_1^a, \ldots, b_k^a$, *compute* $a^{-1}b^{-1}ab$.

We recall a classic definition.

Definition 6 *Let $S \subseteq \mathrm{M}_n(\mathbb{F})$ be a set. The* centralizer *of S (in $\mathrm{M}_n(\mathbb{F})$) is the set*

$$C(S) = \{c \in \mathrm{M}_n(\mathbb{F}) : cs = sc \text{ for all } s \in S\}.$$

For $a_1, \ldots, a_k \in \mathrm{M}_n(\mathbb{F})$, $C(\{a_1, \ldots, a_k\})$ is also denoted as $C(a_1, \ldots, a_k)$.

Basic properties of $C(S)$, that are easy to verify, include:

1. $C(S)$ is a vector subspace (indeed, a matrix subalgebra) of $\mathrm{M}_n(\mathbb{F})$.
2. $C(C(S)) \supseteq S$.
3. $C(S) = C(\operatorname{span} S)$.
4. If $S \subseteq \mathrm{GL}_n(\mathbb{F})$, then $C(S) = C(\langle S \rangle)$, where $\langle S \rangle$ is the subgroup of $\mathrm{GL}_n(\mathbb{F})$ generated by S.

A key observation is the following one: Let V be a vector subspace of $\mathrm{M}_n(\mathbb{F})$, and $G \leq \mathrm{GL}_n(\mathbb{F})$ be a matrix group such that $V \cap G$ is nonempty. It may be computationally infeasible to find an element in $V \cap G$. However, it is easy to compute a basis for $V \cap U$ for any vector subspace U of $\mathrm{M}_n(\mathbb{F})$. In particular, this is true for $U = C(C(G))$, that contains G. In certain cases, as the ones below, a "random" element in $V \cap C(C(G))$ is as good as one in $V \cap G$.

Following is an algorithm for the Commutator KEP Problem in a matrix group $G \leq \mathrm{GL}_n(\mathbb{F})$. The analysis of this algorithm is based on the forthcoming Lemma 9. To this end, we assume that $|\mathbb{F}|/n \geq c > 1$ for some constant c. In the above section, $|\mathbb{F}|/n$ is at least exponential. Fix a finite set $S \subseteq \mathbb{F}$ of cardinality greater than cn (the larger the better), that can be sampled efficiently. In the most important case, where $\mathbb{F}$ is a finite field, take $S = \mathbb{F}$. By *random element* of a vector subspace V of $\mathrm{M}_n(\mathbb{F})$, with a prescribed basis $\{v_1, \ldots, v_d\}$, we mean a linear combination

$$\alpha_1 v_1 + \cdots + \alpha_k v_k$$

with $\alpha_1, \ldots, \alpha_k \in S$ uniform, independently distributed.

It is natural to split the Commutator KEP Problem and the algorithm for solving it into an offline (preprocessing) phase and an online phase.

Algorithm 7
Offline phase:

1. Input: $b_1, \ldots, b_k \in G$.
2. Execution:
   (a) Compute a basis $S = \{s_1, \ldots, s_d\}$ for $C(b_1, \ldots, b_k)$, by solving the following homogeneous system of linear equations in the n^2 entries of the unknown matrix x:

$$\begin{aligned} b_1 \cdot x &= x \cdot b_1 \\ &\vdots \\ b_k \cdot x &= x \cdot b_k. \end{aligned}$$

   (b) Compute a basis for $C(S) = C(C(b_1, \ldots, b_k))$, by solving the following homogeneous system of linear equations in the n^2 entries of the unknown matrix x:

$$\begin{aligned} s_1 \cdot x &= x \cdot s_1 \\ &\vdots \\ s_d \cdot x &= x \cdot s_d. \end{aligned}$$

3. Output: *A basis for* $C(C(b_1, \ldots, b_k))$.

Online phase:

1. Input: $a_1, \ldots, a_k, b_1, \ldots, b_k, a_1^b, \ldots, a_k^b, b_1^a, \ldots, b_k^a \in G$, *where* $a \in \langle a_1, \ldots, a_k \rangle, b \in \langle b_1, \ldots, b_k \rangle$ *are unknown.*
2. Execution:
   (a) *Solve the following homogeneous system of linear equations in the* n^2 *entries of the unknown matrix* x*:*
   $$\begin{aligned} b_1 \cdot x &= x \cdot {b_1}^a \\ &\vdots \\ b_k \cdot x &= x \cdot {b_k}^a. \end{aligned}$$
   (b) *Fix a basis for the solution space, and pick random solutions* x *until* x *is invertible.*
   (c) *Solve the following homogeneous system of linear equations in the* n^2 *entries of the unknown matrix* y*:*
   $$\begin{aligned} a_1 \cdot y &= y \cdot {a_1}^b \\ &\vdots \\ a_k \cdot y &= y \cdot {a_k}^b, \end{aligned}$$
   subject to the linear constraint *that* $y \in C(C(b_1, \ldots, b_k))$.
   (d) *Fix a basis for the solution space, and pick random solutions* y *until* y *is invertible.*
   (e) Output: $x^{-1}y^{-1}xy$.

Let ω be the matrix multiplication constant, that is, the minimal such that matrix multiplication is $O(n^{\omega+o(1)})$. For our applications, one may take $\omega = \log_2 7 \approx 2.81$. As usual, *Las Vegas algorithm* means an algorithm that always outputs the correct answer in finite time. For the proof of the following theorem, note that if $g^x = g^y$, then $g^{xy^{-1}} = g$, or in other words, xy^{-1} commutes with g.

Theorem 8 *Assume that* $|\mathbb{F}|/n \geq c > 1$ *for some constant* c*, and* $k \leq n^2$*. Algorithm 7 is a Las Vegas algorithm for the Commutator KEP Problem, with running time, in units of field operations:*

1. Offline phase: $O(n^{2\omega+2})$.
2. Online phase: $O(kn^{2\omega})$.

Proof. We use the notation of Algorithm 7. First, assume that the algorithm terminates. We prove that its output is $a^{-1}b^{-1}ab$.

$$x^{-1}y^{-1}xy = x^{-1}y^{-1}(xa^{-1})ay.$$

The equations (2)(a) in the online phase of Algorithm 7 assert that ${b_i}^x = {b_i}^a$ for all $i = 1, \ldots, k$. Thus, $xa^{-1} \in C(b_1, \ldots, b_k)$. As $y \in C(C(b_1, \ldots, b_k))$, y commutes with xa^{-1}, and therefore so does y^{-1}. Thus,

$$x^{-1}y^{-1}(xa^{-1})ay = x^{-1}(xa^{-1})y^{-1}ay = a^{-1}y^{-1}ay = a^{-1}a^y.$$

By the equations (2)(c) in the online phase of Algorithm 7, ${a_i}^y = {a_i}^b$ for all $i = 1, \ldots, k$. As $a \in \langle a_1, \ldots, a_k \rangle$, we have that $a^y = a^b$. Indeed, let $a = a_{i_1}^{\epsilon_1} \cdots a_{i_m}^{\epsilon_m}$. As conjugation is an isomorphism,

$$a^y = (a_{i_1}^{\epsilon_1})^y \cdots (a_{i_m}^{\epsilon_m})^y = (a_{i_1}^y)^{\epsilon_1} \cdots (a_{i_m}^y)^{\epsilon_m} = (a_{i_1}^b)^{\epsilon_1} \cdots (a_{i_m}^b)^{\epsilon_m} = (a_{i_1}^{\epsilon_1})^b \cdots (a_{i_m}^{\epsilon_m})^b = a^b.$$

Thus,

$$a^{-1}a^y = a^{-1}a^b = a^{-1}b^{-1}ab.$$

Running time, offline phase: (2)(a) These are kn^2 equations in n^2 variables, and thus the running time is $O(k(n^2)^\omega) = O(kn^{2\omega})$.

(2)(b) As $C(b_1, \ldots, b_k)$ is a vector subspace of $\mathrm{M}_n(\mathbb{F})$, its dimension d is at most n^2. Thus, the running time of this step is $O(n^2 \cdot n^{2\omega}) = O(n^{2\omega+2})$.

Running time, online phase: (2)(a) As in (2)(a) of the offline phase.

(2)(b) There is an invertible solution to the equations (2)(a), namely: a. Thus, by the Invertibility Lemma 9, the probability that a random solution is *not* invertible may be assumed arbitrarily close to $n/|\mathbb{F}| \leq 1/c < 1$. Thus, the expected number of random elements picked until an invertible one was found is constant. To generate one random element, one takes a linear combination of a basis of the solution space. If d is the dimension, then $d \leq n^2$ and the linear combination takes $dn^2 \leq n^4$ operations. Checking invertibility is faster. The total expected running time of this step is, therefore, $O(n^4)$, and $n^4 \leq n^{2\omega}$.

(2)(c) Recycling notation, let $\{s_1, \ldots, s_d\}$ be the basis computed in the offline phase. Then $d \leq n^2$. In the present step, one sets $y = t_1 s_1 + \cdots + t_d s_d$, with $t_1, \ldots, t_d$ variables, and obtains kn^2 equations in the $d \leq n^2$ variables $t_1, \ldots, t_d$. The complexity is $O(\frac{kn^2}{d} d^\omega)$, and $\frac{kn^2}{d} d^\omega = kn^2 \cdot d^{\omega-1} \leq kn^{2\omega}$.

(2)(d) Similar to (2)(b).

6 Finding an invertible solution when there is one

The results in the previous section assume that we are able to find, efficiently, an invertible matrix in any subspace of $\mathrm{M}_n(\mathbb{F})$ containing an invertible element. This is taken care of by the following Lemma.

Lemma 9 (Invertibility Lemma) *Let $a_1, \ldots, a_m \in \mathrm{M}_n(\mathbb{F})$ be such that*

$$\mathrm{span}\{a_1, \ldots, a_m\} \cap \mathrm{GL}_n(\mathbb{F}) \neq \emptyset.$$

Let S be a finite subset of $\mathbb{F}$. If $\alpha_1, \ldots, \alpha_m$ are chosen uniformly and independently from S, then the probability that $\alpha_1 a_1 + \cdots + \alpha_m a_m$ is invertible is at least $1 - \frac{n}{|S|}$.

Proof. Let

$$f(t_1, \ldots, t_m) = \det(t_1 a_1 + \cdots + t_m a_m) \in \mathbb{F}[t_1, \ldots, t_m],$$

where $t_1, \ldots, t_m$ are scalar variables. This is a determinant of a matrix whose coefficients are linear in the variables. By the definition of determinant as a sum of products of n elements, f is a polynomial of degree n. As $\mathrm{span}\{a_1, \ldots, a_m\} \cap \mathrm{GL}_n(\mathbb{F}) \neq \emptyset$, f is nonzero. Apply the Schwartz–Zippel Lemma 10.

For the reader's convenience, we include a proof for the following classic lemma.

Lemma 10 (Schwartz–Zippel) *Let $f(t_1, \ldots, t_m) \in \mathbb{F}[t_1, \ldots, t_m]$ be a nonzero multivariate polynomial of degree n. Let S be a finite subset of $\mathbb{F}$. If $\alpha_1, \ldots, \alpha_m$ are chosen uniformly and independently from S, then the probability that $f(\alpha_1, \ldots, \alpha_m) \neq 0$ is at least $1 - \frac{n}{|S|}$.*

Proof. By induction on m. If $m = 1$, then f is a univariate polynomial of degree n, and thus has at most n roots.

$m > 1$: Write

$$f(t_1, \ldots, t_m) = f_0(t_2, \ldots, t_m) + f_1(t_2, \ldots, t_m)t_1 + f_2(t_2, \ldots, t_m)t_1^2 + \cdots + f_k(t_2, \ldots, t_m)t_1^k,$$

with $k \leq n$ maximal such that $f_k(t_2, \ldots, t_m)$ is nonzero. The degree of $f_k(t_2, \ldots, t_k)$ is at most $m - k$. For each choice of $\alpha_2, \ldots, \alpha_m \in \mathbb{F}$ with $f_k(\alpha_2, \ldots, \alpha_m) \neq 0$, $f(t_1, \alpha_2, \ldots, \alpha_m)$ is a univariate polynomial of degree k in the variable t_1. By the induction hypothesis (for $m = 1$), for random $\alpha_1 \in S$, $f(\alpha_1, \alpha_2, \ldots, \alpha_m)$ is nonzero with probability at least $1 - k/|S|$. By the induction hypothesis,

$$\begin{aligned}
&\Pr\big[f(\alpha_1, \ldots, \alpha_m) \neq 0\big] \geq \\
&\geq \Pr\big[f_k(\alpha_2, \ldots, \alpha_m) \neq 0\big] \cdot \Pr\big[f(\alpha_1, \ldots, \alpha_m) \neq 0 \mid f_k(\alpha_2, \ldots, \alpha_m) \neq 0\big] \geq \\
&\geq \left(1 - \frac{n-k}{|S|}\right)\left(1 - \frac{k}{|S|}\right) \geq 1 - \frac{n}{|S|}.
\end{aligned}$$

7 Application to the Centralizer KEP

Definition 11 *For a group G and an element $g \in G$, the* centralizer of g in G *is the set*

$$C_G(g) := \{h \in G \,:\, gh = hg\}.$$

The *Centralizer KEP*, introduced by Shpilrain and Ushakov in 2006 [33], is described in Figure 2. In this protocol, a_1 commutes with b_1 and a_2 commutes with b_2. Consequently, the keys computed by Alice and Bob are identical, and equal to $a_1 b_1 g a_2 b_2$.

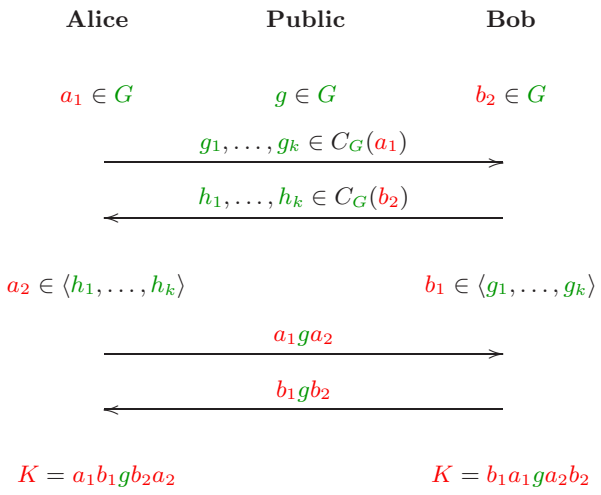

Fig. 2. The Centralizer KEP

As in the Commutator KEP, it is proposed in [33] to use the braid group $\mathbf{B}_N$ as the platform group G. The group elements are chosen in a special way, so as to foil attacks attempted at earlier braid group based KEPs. We apply the methods developed in the previous sections to obtain a polynomial time cryptanalysis of this KEP. We omit some details, that are similar to those in the earlier sections.

Problem 12 (Centralizer KEP Problem) *Assume that* $g, a_1, b_2 \in \mathbf{B}_N$, $g_1, \ldots, g_k \in C_{\mathbf{B}_N}(a_1)$, $h_1, \ldots, h_k \in C_{\mathbf{B}_N}(b_2)$, *each of the form* $(i, \mathbf{p})$ *with* $\mathbf{p}$ *of length* $\leq \ell$. *Let* a_2 *be a product of at most* m *elements of* $\{h_1, \ldots, h_k\}^{\pm 1}$, *and let* b_1 *be a product of at most* m *elements of* $\{g_1, \ldots, g_k\}^{\pm 1}$.
Given $g, g_1, \ldots, g_k, h_1, \ldots, h_k, a_1 g a_2, b_1 g b_2$, *compute* $a_1 b_1 g a_2 b_2$.

7.1 Solving the Centralizer KEP Problem in matrix groups.

For a group G, $Z(G) = C_G(G)$ is the set of all central elements of G. Consider the Centralizer KEP Problem 12 in $G \leq \mathrm{GL}_n(\mathbb{F})$ instead of $\mathbf{B}_N$. The following variation of this problem is formally harder.

Problem 13 *Let* $G \leq \mathrm{GL}_n(\mathbb{F})$. *Assume that* $g, a_1, b_2 \in G$, $g_1, \ldots, g_k \in C_G(a_1)$, $h_1, \ldots, h_k \in C_G(b_2)$, $a_2 \in \langle \{h_1, \ldots, h_k\} \cup Z(G) \rangle$, *and* $b_1 \in \langle \{g_1, \ldots, g_k\} \cup Z(G) \rangle$.
Given $g, g_1, \ldots, g_k, h_1, \ldots, h_k, a_1 g a_2, b_1 g b_2$, *compute* $a_1 b_1 g a_2 b_2$.

Following is an algorithm for solving it. As before, for $S \subseteq \mathrm{M}_n(\mathbb{F})$, $C(S)$ (without subscript) is the centralizer of S *in the matrix algebra* $\mathrm{M}_n(\mathbb{F})$.

Algorithm 14

1. Input: $g, g_1, \ldots, g_k, h_1, \ldots, h_k, a_1 g a_2, b_1 g b_2 \in G$.
2. Execution:
   (a) *Compute bases for the subspaces* $C(g_1, \ldots, g_k)$, $C(C(h_1, \ldots, h_k))$ *of* $\mathrm{M}_n(\mathbb{F})$.
   (b) *Solve*
   $$x \cdot g = a_1 g a_2 \cdot y$$
   subject to the linear constraints $x \in C(g_1, \ldots, g_k), y \in C(C(h_1, \ldots, h_k))$.
   (c) *Take random linear combinations of the basis of the solution space to obtain solutions* (x, y), *until* y *is invertible.*
3. Output: $x \cdot b_1 g b_2 \cdot y^{-1}$.

Theorem 15 *Let* $G \leq \mathrm{GL}_n(\mathbb{F})$. *Assume that* $|\mathbb{F}|/n \geq c > 1$ *for some constant* c, *and* $k \leq n^2$. *Algorithm 14 is a Las Vegas algorithm for Problem 13, with running time, in units of field operations,* $O(n^{2\omega+2})$.

Proof. The proof is similar to that of Theorem 8.

First, assume that the algorithm terminates. We prove that its output is $a_1 b_1 g a_2 b_2$. As $x \in C(g_1, \ldots, g_k)$ and $b_1 \in \langle g_1, \ldots, g_k \rangle$, x commutes with b_1. As b_2 commutes with $h_1, \ldots, h_k$, $b_2 \in C(h_1, \ldots, h_k)$. As $y \in C(C(h_1, \ldots, h_k))$, y commutes with b_2, and therefore so does y^{-1}. Thus,

$$x b_1 g b_2 y^{-1} = b_1 x g y^{-1} b_2.$$

As $xg = a_1 g a_2 y$, $xgy^{-1} = a_1 g a_2$. Thus,

$$b_1 x g y^{-1} b_2 = b_1 a_1 g a_2 b_2 = a_1 b_1 g a_2 b_2.$$

Running time: (2)(a) We are solving kn^2 equations in n^2 variables, and then at most n^2 equations in n^2 variables. This is $O(n^{2\omega+2})$.

(2)(b) We are solving n^2 equations in $2n^2$ variables, which is $O(n^{2\omega})$.

(2)(c) Let

$$H = \{(x, y) \in C(g_1, \ldots, g_k) \times C(C(h_1, \ldots, h_k)) \,:\, x \cdot g = a_1 g a_2 \cdot y\}$$

be the solution space, and let $(x_1, y_1), \ldots, (x_d, y_d)$ be a basis for H. As H is a subspace of $\mathrm{M}_n(\mathbb{F}) \times \mathrm{M}_n(\mathbb{F})$, $d \leq 2n^2$. Let $H_2 = \{y \,:\, (x, y) \in H\}$, the projection of H on the second coordinate. Then

$$H_2 = \mathrm{span}\{y_1, \ldots, y_d\}.$$

$(a_1, a_2^{-1}) \in H$, and thus $a_2^{-1} \in H_2$. In particular, there is an invertible element in H_2. By the Invertibility Lemma 9, a random linear combination of $y_1, \ldots, y_d$ is invertible with probability at least $1/c$. The total expected running time of this step is, therefore, $O(n^4)$, and $n^4 \leq n^{2\omega}$.

7.2 Infimum reduction.

In Section 3, we explained how each $x \in \mathbf{B}_N$ can be decomposed (in linear time) as $x = c\tilde{x}$ with c central and $\inf(x) \in \{0, 1\}$.

We may assume that

$$\inf(g) \in \{0, 1\}.$$

Indeed, assume that we have an algorithm solving the problem when $\inf(g) \in \{0, 1\}$. Write $g = c\tilde{g}$ with c central and $\inf(g) \in \{0, 1\}$. Compute

$$c^{-1} a_1 g a_2 = a_1 c^{-1} g a_2 = a_1 \tilde{g} a_2;$$
$$c^{-1} b_1 g b_2 = b_1 c^{-1} g b_2 = b_1 \tilde{g} b_2.$$

Apply the given algorithm to $\tilde{g}, g_1, \ldots, g_k, h_1, \ldots, h_k, a_1\tilde{g}a_2, b_1\tilde{g}b_2$, to obtain $a_1 b_1 \tilde{g} a_2 b_2$. Multiply by c to obtain $a_1 b_1 g a_2 b_2$.

Next, we may assume that

$$\inf(g_1), \ldots, \inf(g_k), \inf(h_1), \ldots, \inf(h_k) \in \{0, 1\},$$

since when we apply Algorithm 14 in the image of our group in a matrix group, we have in Problem 13 that

$$\langle \{h_1, \ldots, h_k\} \cup Z(G) \rangle = \langle \{\tilde{h}_1, \ldots, \tilde{h}_k\} \cup Z(G) \rangle;$$
$$\langle \{g_1, \ldots, g_k\} \cup Z(G) \rangle = \langle \{\tilde{g}_1, \ldots, \tilde{g}_k\} \cup Z(G) \rangle.$$

As in Section 3, it follows that

$$a_2, b_1 \in [-m(\ell + 1), m(\ell + 1)].$$

Let $u = a_1 g a_2$ and $v = b_1 g b_2$. Decompose $u = c\tilde{u}$ and $v = d\tilde{v}$ with c, d central and $\inf(\tilde{u}), \inf(\tilde{v}) \in \{0, 1\}$. As $g \in [0, \ell + 1]$ and $a_1 \in [\inf(a_1), \inf(a_1) + \ell]$,

$$u = a_1 g a_2 \in [\inf(a_1), \inf(a_1) + (m + 1)(\ell + 1) + \ell],$$

and thus

$$a_1 g (c^{-1} a_2) = \tilde{u} \in [0, (m + 1)(\ell + 2)];$$
$$c^{-1} a_1 = \tilde{u} a_2^{-1} g^{-1} \in [-(m + 1)(\ell + 1), (m + 1)(2\ell + 3)].$$

Similarly,

$$(d^{-1}b_1)gb_2 = \tilde{v} \in [0, (m+1)(\ell+2)].$$

Finally,

$$K' := a_1(d^{-1}b_1)gb_2(c^{-1}a_2) = a_1\tilde{v}(c^{-1}a_2) = (c^{-1}a_1)\tilde{v}a_2 \in [-(m+2)(\ell+1), (m+1)(4\ell+6)].$$

Let $M = (m+2)(4\ell+6)$. Continue as in Section 3.

By Theorem 4, we have that

$$(2^{2NM}t^M) \cdot \mathrm{LK}(K') \in \mathrm{GL}_n(\mathbb{Z}[t]),$$

the absolute values of the coefficients in this matrix are bounded by $2^{N^2(M+1)}$, and the maximal degree of t in this matrix is bounded by $2M$. Let p be a prime slightly greater than 2^{N^2M}, and $f(t)$ be an irreducible polynomial over $\mathbb{Z}_p$, of degree d slightly larger than $2M$. Then

$$(2^{2NM}t^M) \cdot \mathrm{LK}(K') = (2^{2NM}t^M) \cdot \mathrm{LK}(K') \bmod (p, f(t)) \in \mathrm{GL}_n(\mathbb{Z}[t]/\langle p, f(t)\rangle),$$

under the natural identification of $\{-(p-1)/2, \ldots, (p-1)/2\}$ with $\{0, \ldots, p-1\}$. Let $\mathbb{F} = \mathbb{Z}[t]/\langle p, f(t)\rangle = \mathbb{Z}[t^{\pm 1}, \frac{1}{2}]/\langle p, f(t)\rangle$. $\mathbb{F}$ is a finite field of cardinality p^d, where d is the degree of $f(t)$. It follows that the complexity of field operations in $\mathbb{F}$ is, up to logarithmic factors, of order

$$d^2 \log p = O(M^3N^2) = O(m^3\ell^3N^2).$$

Thus, the key K can be recovered as follows:

1. Apply the composed function $\mathrm{LK}(x) \bmod (p, f(t))$ to $g, g_1, \ldots, g_k, h_1, \ldots, h_k, \tilde{u} = a_1g(c^{-1}a_2), \tilde{v} = (d^{-1}b_1)gb_2$, to obtain an input to Problem 13.
2. Solve the problem there, to obtain $\mathrm{LK}(K') \bmod (p, f(t))$.
3. Compute $(2^{2NM}t^M) \cdot \mathrm{LK}(K') \bmod (p, f(t)) = (2^{2NM}t^M) \cdot \mathrm{LK}(K')$.
4. Divide by $(2^{2NM}t^M)$ to obtain $\mathrm{LK}(K')$.
5. Compute K' using the Cheon–Jun inversion algorithm.
6. Multiply by cd to obtain $a_1b_1ga_2b_2$.

8 Final comments

Ignoring logarithmic factors, the overall complexity of both cryptanalyses presented here is $n^{2\omega+2} = N^{4\omega+4}$ field operations, that are of complexity $m^3\ell^3N^2$. Thus, the complexity is

$$N^{4\omega+6}m^3\ell^3,$$

ignoring logarithmic factors. While polynomial, this complexity practical only for braid groups of small index N. However, even the problems of finding polynomial time attacks on the Commutator KEP or on the Centralizer KEP were open up till now.

The methods introduced here are also applicable for cryptanalyzing other KEPs. For example, the Invertibility Lemma can be used to turn both the Cheon–Jun cryptanalysis of the Braid Diffie–Hellman KEP [6] and the Shpilrain cryptanalysis of Stickel's KEP [31] into Las Vegas algorithms of expected polynomial time. Our infimum reductions can be applied

to the Cheon–Jun attack to eliminate the exponential dependence on the bitlength of the infimum, a technical issue that was apparently not treated thus far.

The major challenge is to reduce the degree of N in the polynomial time cryptanalyses. By Chinese Remaindering or p-adic lifting methods, it may be possible to reduce the complexity contributed by the field operations. Apparently, this may reduce the power of N by 1. It should be possible to make sure that the Invertibility Lemma is still applicable when these methods are used. Much of the complexity comes from the Lawrence–Krammer representation having dimension quadratic in N. It may well be that there are no faithful representations of $\mathbf{B}_N$ of smaller dimension. Finally, a more careful analysis of the Lawrence–Krammer representation may make it possible to obtain finer estimates. However, it does not seem that any of these directions would make the attacks practical for, say, $N = 100$.

One may wonder whether, from the CS *theory* point of view, this paper may end up braid-based cryptography. My belief is that this is not the csae. I do not at present know whether Kurt's *Triple Decomposition KEP* [26, 4.2.5] can be cryptanalyzed using the methods presented here. Additional KEPs to which the present methods do not seem to be applicable are introduced by Kalka in [17] and [18]. Moreover, there are additional types of braid-based schemes (e.g., authentication schemes), that cannot be attacked using the methods presented here. Some examples are reviewed in the monograph [26].

Changing the platform group in any of the studied KEPs is a very interesting option. There are efficiently implementable, infinite groups with no faithful representations as matrix groups. As for finite groups, I am pessimistic. For example, finite simple groups tend to be linear, by the classification of finite simple groups, and our method would reduce the cryptanalysis to the problem of finding an *efficient* linear representation. There are at present no signs that such representations must be harder to evaluate (or invert) than, say, solving the discrete logarithm problem in $\mathbb{Z}_p^*$.

Acknowledgments

I worked on the Commutator KEP, from various other angles, since I was introduced to it at the Hebrew University CS Theory seminar, by Alex Lubotzky [30]. I thank Oleg Bogopolski for inviting me, earlier this year (2012), to deliver a minicourse in the conference *Geometric and Combinatorial Group Theory with Applications* (Düsseldorf, Germany, July 25–August 3, 2012). Preparing this minicourse, I discovered the linear centralizer attack. Initially, I addressed the Centralizer KEP (Section 7). When I moved to consider the Commutator KEP, Arkadius Kalka pointed out an obstacle mentioned by Shpilrain and Ushakov, that stroke me as solvable by linear centralizers. I am indebted to Kalka for making the right comment at the right time.

I also thank David Garber, Arkadius Kalka, and Eliav Levy, for comments leading to improvements in the presentation of this paper.

References

1. B. An, K. Ko, *A family of pseudo-Anosov braids with large conjugacy invariant sets*, ArXiv eprint arXiv:1203.2320, 2012.
2. I. Anshel, M. Anshel, D. Goldfeld, *An algebraic method for public-key cryptography*, Mathematical Research Letters **6** (1999), 287–291.
3. I. Anshel, M. Anshel, B. Fisher, D. Goldfeld, *New Key Agreement Protocols in Braid Group Cryptography*, CT-RSA 2001, Lecture Notes in Computer Science **2020** (2001), 13–27.

4. S. Bigelow, *Braid groups are linear*, Journal of the American Mathematical Society **14** (2001), 471–486.
5. J. Cha, K. Ko, S. Lee, J. Han, J. Cheon, *An efficient implementation of braid groups*, ASIACRYPT 2001, LNCS **2248** (2001), 144–156.
6. J. Cheon, B. Jun, *A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem*, CRYPTO 2003, Lecture Notes in Computer Science **2729** (2003), 212–224.
7. P. Dehornoy, *Braid-based cryptography*, Contemporary Mathematics **360** (2004), 5–33.
8. D. Garber, *Braid group cryptography*, in: J. Berrick, F.R. Cohen, E. Hanbury, Y.L. Wong, J. Wu, eds., **Braids: Introductory Lectures on Braids, Configurations and Their Applications**, IMS Lecture Notes Series **19**, National University of Singapore, 2009, 329–403.
9. D. Garber, S. Kaplan, M. Teicher, B. Tsaban, U. Vishne, *Probabilistic solutions of equations in the braid group*, Advances in Applied Mathematics **35** (2005), 323–334.
10. V. Gebhardt, *A new approach to the conjugacy problem in Garside groups*, Journal of Algebra **292** (2005), 282–302.
11. V. Gebhardt, *Conjugacy search in braid groups*, Applicable Algebra in Engineering, Communication and Computing **17** (2006), 219–238.
12. R. Gilman, A. Miasnikov, A. Miasnikov, A. Ushakov, *New developments in Commutator Key Exchange*, Proceedings of the First International Conference on Symbolic Computation and Cryptography, Beijing, 2008, 146–150. `http://www-calfor.lip6.fr/ jcf/Papers/scc08.pdf`
13. D. Hofheinz, R. Steinwandt, *A practical attack on some braid group based cryptographic primitives*, PKC 2003, Lecture Notes in Computer Science **2567** (2002), 187–198.
14. J. Hughes, A. Tannenbaum, *Length-based attacks for certain group based encryption rewriting systems*, SECI02: Sécurité de la Communication sur Internet, 2002. `www.ima.umn.edu/preprints/apr2000/1696.pdf`
15. J. Hughes, *A linear algebraic attack on the AAFG1 braid group cryptosystem*, Information Security And Privacy, Lecture Notes in Computer Science **2384** (2002), 107–141.
16. A. Kalka, *Representation attacks on the braid Diffie-Hellman public key encryption*, Applicable Algebra in Engineering, Communication and Computing **17** (2006), 257–266.
17. A. Kalka, *Representations of braid groups and braid-based cryptography*, PhD thesis, Ruhr-Universität Bochum, 2007. `www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/KalkaArkadiusG/`
18. A. Kalka, *Non-associative public key cryptography*, preprint, 2012.
19. K. Ko, S. Lee, J. Cheon, J. Han, J. Kang, C. Park, *New public-key cryptosystem using braid groups*, CRYPTO 2000, Lecture Notes in Computer Science **1880** (2000), 166–183.
20. K. Ko, J. Lee, T. Thomas, *Towards generating secure keys for braid cryptography*, Design Codes and Cryptography **45** (2007), 317–333.
21. D. Krammer, *Braid groups are linear*, Annals of Mathematics **155** (2002), 131–156.
22. S. Lee, E. Lee, *Potential weaknesses of the commutator key agreement protocol based on braid groups*, EUROCRYPT 2002, Lecture Notes in Computer Science **2332** (2002), 14–28.
23. S. Maffre, *A weak key test for braid-based cryptography*, Design Codes and Cryptography **39** (2006), 347–373.
24. A. Miasnikov, V. Shpilrain, A. Ushakov, *A practical attack on some braid group based cryptographic protocols*, CRYPTO 2005, Lecture Notes in Computer Science **3621** (2005), 86–96.
25. A. Miasnikov, V. Shpilrain, A. Ushakov, *Random subgroups of braid groups: an approach to cryptanalysis of a braid group based cryptographic protocol*, PKC 2006, Lecture Notes in Computer Science **3958** (2006), 302–314.
26. A. Miasnikov, V. Shpilrain, A. Ushakov, **Non-commutative Cryptography and Complexity of Group-theoretic Problems**, American Mathematical Society Surveys and Monographs **177**, 2011.
27. A. Miasnikov, A. Ushakov, *Length based attack and braid groups: cryptanalysis of Anshel-Anshel-Goldfeld key exchange protocol*, PKC 2007, Lecture Notes in Computer Science **4450** (2007), 76–88.
28. A. Myasnikov, A. Ushakov, *Random subgroups and analysis of the length-based and quotient attacks*, Journal of Mathematical Cryptology **2** (2008), 29–61.
29. D. Micciancio, O. Regev, *Lattice-based Cryptography*, in: **Post-quantum Cryptography** (D. Bernstein and J. Buchmann, eds.), Springer, 2008.
30. A. Lubotzky, *Braid group cryptography*, CS Theory Seminar, Hebrew University, March 2001. `http://www.cs.huji.ac.il/ theorys/2001/Alex_Lubotzky`
31. V. Shpilrain, *Cryptanalysis of Stickel's key exchange scheme*, in: **Computer Science in Russia**, Lecture Notes in Computer Science 5010 (2008), 283–288.
32. V. Shpilrain, A. Ushakov, *Thompson's group and public key cryptography*, ACNS 2005, Lecture Notes in Computer Science **3531** (2005), 151–164.
33. V. Shpilrain, A. Ushakov, *A new key exchange protocol besed on the decomposition problem*, in: L. Gerritzen, D. Goldfeld, M. Kreuzer, G. Rosenberger and V. Shpilrain, eds., **Algebraic Methods in Cryptography**, Contemporary Mathematics **418** (2006), 161–167.